Identifier

titanId

Within Titan, each and every single item has an Id - simply an integer number which is this item's unique identifier. This can be used to refer to a specific item by calling this object with its id.

Usually, each id is set by Titan itself - you have no means - and no reason - to alter a particular item's id. Also, that's why **you will use the titanId in macros only in rare cases: there is no practical way to find the titanId of a specific item, and it might be different in the next show.** However, there are some reserved IDs which might be a good guess for the master IDs in new shows. **As you can see below this is not guaranteed, in particular across versions – the IDs have changed from v10 to v11**:

Titan Id v10	Titan Id v11	Handle	Description
1600	1605	GrandMaster	Grand Master
1601	1606	FlashMaster	Flash Master
1602	1607	PlaybackMaster	PlaybackMaster
1603	1608	PresetMaster	Preset Master
1604	1609	RateGrandMaster	Rate Grand Master
1605	1610	SwopMaster	Swop Master
1606	1611	SizeGrandMaster	Size Grand Master
1607	1612	BPMMaster:0	BPM Master 1
1608	1613	RateMaster:0	Rate Master 1
1609	1614	SizeMaster:0	Size Master 1
1610	1615	(empty)	
1611	1616	BPMMaster:1	BPM Master 2
1612	1617	RateMaster:1	Rate Master 2
1613	1618	SizeMaster:1	Size Master 2
1614	1619	(empty)	
1615	1620	BPMMaster:2	BPM Master 3
1616	1621	RateMaster:2	Rate Master 3
1617	1622	SizeMaster:2	Size Master 3
1618	1623	(empty)	
1619	1624	BPMMaster:3	BPM Master 4
1620	1625	RateMaster:3	Rate Master 4
1621	1626	SizeMaster:3	Size Master 4
1622	1627	(empty)	
1623	1628	ABMaster	A/B Master

(found by scrutinizing showfiles after uncompressing)

There is a website which retrieves titanIds from your show if called on the same computer where Titan PC suite (and Web API) are running: http://www.avolites.de/downloads/miditoweb/gettitanids.htm

ASSUMPTION (derived from some examples)

> > In order to call an item by its id, you simply pass its titanId as handle, without any other identifiers:

```
<!-- function definition:
  Void Masters.TapTempo(Handle handle, DateTime panelTimeStamp)
-->

<step>Masters.TapTempo(1607, Math.GetCurrentTimeStamp())</step>
```

Further explanation from http://forum.avolites.com/viewtopic.php?f=20&t=5731#p20666:

> For a given show file the IDs will remain the same once they are allocated. The IDs may have seemed fixed however this was only because they are usually about the first things to be allocated so might often end up with the same numbers (1600 is the start of the non-fixed IDs). There was an intention when rate and BPM masters were added that at some point we would likely want to allow for any number of masters to be created so for this reason the software was written to allocate them dynamically. These masters were added several versions before WebAPI and macro documation was released so this wasn't a consideration at the time. Perhaps a way can be added which can allow you to reference masters properly and reliably in macros in future versions.
>
> When importing, unless you are mapping to something that already exists in the current show, new Titan IDs will be allocated; this prevents IDs from clashing.

Also used in

- Masters - BPM Master (snippet)
- Masters - Nudge Master Up/Down
- Masters - Rate Master Functions
- Masters - Reset and Set
- Set Grandmaster
- Masters - Size Master Functions
- Masters - BPM - Tap
- Masters - BPM Master - Set Speed
- Masters.DeadBlackOut
- Masters.DoubleOrHalfSpeedMultiplier
- Masters.NudgeDown
- Masters.NudgeUp
- Masters.ResetSpeedMultiplier
- Masters.SetMaster
- Masters.SetSpeed
- Timelines.ImportMarkersFromString
- Handle
- userNumber
- Identifiers
- Properties list
- MIDI To Web
- Handle
- Masters/ClearFlash
- Masters/DeadBlackOut

- Masters/Flash
- Masters/SetMaster
- Masters/SetSpeed